# HyperText Transfer Protocol

Protocolo de Transferência de HyperTexto.

## Visão Geral:

* Permite troca de informação e dados na internet;
* Uma troca de mensagens;
* HTML, CSS, Scripts, Imagens, Vídeo, ...;
* Uma chamada para cada um desses recursos.

## Conceitos:

* Simples:
  + Legível;
  + Qualquer pessoa;
* Cliente/ Servidor:
  + Requisição/Resposta;
  + Pedir um lanche;
* Stateless:
  + Não guarda informações;
  + Não existe relação entre conexões;
  + Jogar uma moeda;
  + Sessões:
    - Cookies;
    - Storages;
* Extensível:
  + Através do cabeçalho, podemos fazer diversas trocas de informações entre o cliente-servidor, conforme a necessidade;
  + Headers:
    - Informações para comunicação;
  + Body:
    - Corpo do pedido ou da resposta;

Cliente:

* Quem é o cliente?
  + User Agent:
    - Browser;
    - cURL;
  + Entidade que dá início à comunicação;
* Pedidos:
  + Ações do cliente:
    - GET;
    - POST;
    - PUT;
    - DELETE;

Servidor:

* Se apresenta como uma máquina;
* Preparado para ouvir e processar;
* Vários servidores em um computador;
* Resposta:
  + Headers:
    - Status Code:
      * 404 – Não encontrado;
      * 500 – Erro no servidor;

Proxies:

* Representantes;
* Fica entre o cliente e o servidor;
* Ajudam a fazer o transporte dos dados;
* Diversas funções:
  + Cache;
  + Filtro (tipo um antivírus, ou um controle parental);
  + Load balancing (distribuição da carga);
  + Autenticação;
  + Autorização;

## URI

Conceito:

* Uniform Resource Identifier:
  + Identificar um recurso;
  + Nome e localização;
  + Exemplo:
    - Você é um recurso;
    - Seu nome;
    - Sua localização;

Resource:

* O alvo do pedido;
* Qualquer coisa identificável / Entidade:
  + Digital:
    - E-mail:
      * Mailto: email@dominio.com;
  + Abstrata:
    - Sessão;
    - Autenticação;
  + Física:
    - Produtos;
    - Usuários;
  + Se podemos identificar, nomear, endereçar ou manipular, estamos falando de um recurso.

Locator:

* URL:
  + Uniform Resource Locator;
  + Componentes:
    - Obrigatórios:
      * Protocolo;
      * Domínio;
    - Opcionais:
      * Subdomínio;
      * Path;
      * Parâmetros;
      * Porta;
      * Ancora;
  + Exemplos:
    - https://www.rocketseat.com.br/blog
    - https://www.youtube.com/watch?v=vpYct2npKD8
    - http://127.0.0.1:3333/index.html#algumlugar

Name:

* URN
  + Uniform Resource Name;
  + Exemplos:
    - urn:isbn:0451450523
    - urn:oasis:names:specification:docbook:dtd:xml:4.1.2

## Request 🡪 HTTP Messages 🡨 Response

HTTP Messages:

* Mensagens;
* HTTP/1.1:
  + Legível;
  + Texto;
* HTTP/2:
  + Estrutura binária;
  + Otimizações;
  + Mesma maneira que a versão 1.1;

Request:

* Request Line:
  + Method;
  + Protocol version;
  + URI;
* Body;
* Headers;

Response:

* Protocol version;
* Status code;
* Hearders;
* Status message;

## HTTP Methods

* Define um conjunto de métodos HTTP;
* Indica a ação que o cliente deseja operar;
* Podem ser chamadas de Verbos HTTP;
* Cada um possui a sua semântica;
* Características:
  + Seguros:
    - Não altera o estado do servidor;
    - Somente leitura;
    - Cliente não solicita alterações;
    - Não há carga extra para o servidor
    - O servidor é responsável em manter o método seguro;
    - Quais são:
      * GET:
        + GET / search.html HTTP/1.1
      * HEAD;
      * OPTIONS;
  + Idempotentes:
    - Ao executar o método, a resposta deverá ser sempre a mesma;
    - Quais são:
      * Todos os seguros são idempotentes;
      * PUT;
      * DELETE;
    - Status code poderá ser diferente;
    - O servidor tem a responsabilidade de retornar dados da mesma maneira;
    - Essa especificação não é garantia de que todos os servidores irão aplicar o conceito corretamente;